Adaptive Automation and Cue Invocation: The Effect of Cue Timing on Operator Error

Daniel Gartenberg
George Mason University
gartenbergdaniel@gmail.com

Leonard A. Breslow
Naval Research Laboratory
len.breslow@nrl.navy.mil

Joo Park
George Mason University
park.joo@gmail.com

J. Malcolm McCurry
ITT Exelis
malcolm.mccurry.ctr@nrl.navy.mil

J. Gregory Trafton
Naval Research Laboratory
greg.trafton@nrl.navy.mil

ABSTRACT

Adaptive automation (AA) can improve performance while addressing the problems associated with a fully automated system. The best way to invoke AA is unclear, but two ways include critical events and the operator's state. A hybrid model of AA invocation, the dynamic model of operator overload (DMOO), that takes into account critical events and the operator's state was recently shown to improve performance. The DMOO initiates AA using critical events and attention allocation, informed by eye movements. We compared the DMOO with an inaccurate automation invocation system and a system that invoked AA based only on critical events. Fewer errors were made with DMOO than with the inaccurate system. In the critical event condition, where automation was invoked at an earlier point in time, there were more memory and planning errors, while for the DMOO condition, which invocated automation at a later point in time, there were more perceptual errors. These findings provide a framework for reducing specific types of errors through different automation invocation.

Author Keywords

Adaptive Automation; Situation Awareness; Fan-out; Trust in Automation; Eye Tracking; Errors; Supervisory Control.

ACM Classification Keywords

H.5.2, H.1.2.

INTRODUCTION

Human operators are increasingly taking on the role of supervisor in complex semi-autonomous computer systems. Yet increased automation comes with unexpected side effects. This is because automation does not simply supplant human behavior, but rather, it interacts with human behavior in unintended ways [1]. This can include reduced situation awareness, inappropriate trust, unbalanced workload, decision biases, and over-reliance and complacency [2, 3, 4]. As a result, there is a need to understand the types of errors that operators make when interacting with such systems.

Adaptive automation (AA) has been proposed as a solution to the negative consequences of more-typical, static automation [5, 6, 7]. In an adaptive automation system, automation is flexible and responsive to the needs of the user and the changes in the task environment [8]. AA can be invoked by: a) critical events; b) operator performance measures; c) operator physiological assessment; d) operator modeling; and/or e) hybrid methods [9]. While empirical evaluations have shown the efficacy of adaptive automation in such domains as aviation [10], air traffic management [11], and industrial process control [12], the method of invocation can dramatically affect how the operator performs on a given task [13].

Of these methods of invocation, recent evidence points to the potential advantage of a hybrid invocation method to improve operator performance, where the operator's attention, evaluated by analyzing eye movements, and a critical event are used to invoke automation. This was recently demonstrated by the dynamic model of operator overload, which assessed the urgency of the critical event and the operator's attention allocation to predict whether a vehicle will fly into a hazard. When the model predicted an error, a cue was initiated on the situation in question, which resulted in over 50% fewer errors [14].

The dynamic model of operator overload (DMOO) was an adaptation of the fan-out model, initially proposed by Crandall et al. (2005) [15]. Fan-out specifies the maximum number of vehicles that an operator can effectively control by taking into account, *interaction time* (IT) -- how long it takes to interact with a vehicle in order for it to be in an acceptable state, and *neglect time* (NT) -- how long the vehicle can be ignored before it needs attention. Variables that captured cognitive limitations of the operator were later included in the fan-out model, such as wait time due to *human decision making* called interaction wait time (WTI), wait time due to *attention allocation* (WTAA), and wait time due to attending to other vehicles that were ahead in

Report Documentation Page		

1. REPORT DATE **MAY 2013**	2. REPORT TYPE	3. DATES COVERED **00-00-2013 to 00-00-2013**

4. TITLE AND SUBTITLE	5a. CONTRACT NUMBER
Adaptive Automation and Cue Invocation: The Effect of Cue Timing on Operator Error	5b. GRANT NUMBER
	5c. PROGRAM ELEMENT NUMBER
6. AUTHOR(S)	5d. PROJECT NUMBER
	5e. TASK NUMBER
	5f. WORK UNIT NUMBER

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES) **Naval Research Laboratory,4555 Overlook Ave., SW,Washington,DC,20375**	8. PERFORMING ORGANIZATION REPORT NUMBER
9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10. SPONSOR/MONITOR'S ACRONYM(S)
	11. SPONSOR/MONITOR'S REPORT NUMBER(S)

12. DISTRIBUTION/AVAILABILITY STATEMENT
Approved for public release; distribution unlimited

13. SUPPLEMENTARY NOTES
Human Factors in Computing Systems (CHI 2013), April 27?May 2, 2013, Paris, France.

14. ABSTRACT

Adaptive automation (AA) can improve performance while addressing the problems associated with a fully automated system. The best way to invoke AA is unclear, but two ways include critical events and the operator?s state. A hybrid model of AA invocation, the dynamic model of operator overload (DMOO), that takes into account critical events and the operator?s state was recently shown to improve performance. The DMOO initiates AA using critical events and attention allocation, informed by eye movements. We compared the DMOO with an inaccurate automation invocation system and a system that invoked AA based only on critical events. Fewer errors were made with DMOO than with the inaccurate system. In the critical event condition, where automation was invoked at an earlier point in time, there were more memory and planning errors, while for the DMOO condition, which invocated automation at a later point in time, there were more perceptual errors. These findings provide a framework for reducing specific types of errors through different automation invocation.

15. SUBJECT TERMS					
16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES	19a. NAME OF RESPONSIBLE PERSON
a. REPORT **unclassified**	b. ABSTRACT **unclassified**	c. THIS PAGE **unclassified**	**Same as Report (SAR)**	**10**	

the *queue* (WTQ) [16] (see Equation 1).

$$FO = (NT / (IT(WTI) + WTAA + WTQ)) + 1$$

Equation 1: The Fan-out model initially developed by Crandall et al. (2005) and later modified by Cummings et al. (2008) that predicts the maximum number of vehicles that can be simultaneously controlled. WTI is wait time associated with time spent on decision- making and is nested within IT.

The DMOO uses similar theoretical concepts as the fan-out model, yet instead of predicting the maximum number of vehicles that an operator can control, the DMOO predicts operator overload in real-time during the performance of a dynamic task. The model predicts operator overload based on whether or not the operator will respond to a critical event, such as whether or not a vehicle will intersect with a hazardous area. Three theoretical constructs from the fan-out model (see Equation 1) are used as predictors for operator overload in the DMOO. In DMOO, WTAA is instantiated as the amount of time it took to look at objects involved in the situation in question, WTQ is instantiated as the number of fixations on irrelevant objects, and NT is instantiated as the interval of time from when the vehicle's projected path first becomes critical for the operator to address, i.e. when the vehicle intersected a hazardous area, to the predicted moment that the vehicle will make contact with the hazard.

Another consideration when developing the DMOO, and other AA systems, is what component of information processing to automate. Parasuraman, Sheridan, and Wickens (2000) [17] proposed four levels of information processing: sensory processing, perception/working memory, decision-making, and response selection. Each of these four levels can be automated to different degrees where at the lowest end of the spectrum the computer offers no assistance and at the highest end of the spectrum the computer acts autonomously and ignores the human. For the DMOO, decision-making was automated at a medium level on Parasuraman et al.'s (2000) [17] scale, where the automation suggests an alternative action by cueing the operator to a situation needing attention. The reason for automating at this level is that higher levels of automation can have negative consequences, such as reduced situation awareness [11] and increased complacency [18].

Due to the unintended consequences that automation often has on human performance [17], it is important to explore how and why the DMOO is effective. Since increased inaccuracy of automation typically results in worse performance [5, 19], the accuracy of the DMOO should reduce operator errors. However, it is unclear whether there will be similar types of errors based on the accuracy of the automation. Three types of errors were distinguished with respect to how participants respond to an adaptive automation cue: (a) The participant did not look at the situation at all (Noticing Errors), (b) The participant did not look at the situation in time to act (Time-out Errors), (c)

The participant looked at the situation in time, decided to act on something else, and did not act on the cued situation in time (Delay Errors).

Additionally, since the hybrid method invokes automation based on both a critical event in the environment and the operator's state, the relative contributions of these two factors in reducing operator error are unclear. In particular, the timing of the automation is a major difference between a hybrid invocation technique and a critical event invocation technique. If automation is invoked based solely on critical events, the system does not have to take into account the operator's cognitive state, so the cue can be invoked much earlier. Thus, when critical event automation is initiated at the instance that a vehicle needs attention (Point A on Figure 1), the operator quickly becomes aware of the situation, yet may decide to act on something else, since they have more time. But with the DMOO, the automation is invoked later, resulting in an urgent scenario (Point B on Figure 1), whereas the participant is more likely to be responsive to the automation and act on the situation the instant it enters attention.
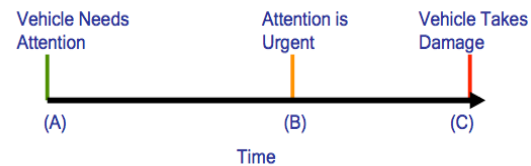

Figure 1: This is an example of a critical event. (A) represents invoking the automation based solely on a critical event, (B) represents invoking the automation using the DMOO (B), and (C) is the negative outcome of damage that occurs when the participant does not respond to the critical event.

Thus, earlier invocation of automation, such as critical event invocation, may result in participants being more likely to look at the automation but decide to act on something else first. A result of looking at other objects after looking at the cue is that prospective memory errors and planning errors are more likely. Prospective memory errors involve memory for intended actions that are planned to be performed at some designated point in the future [20]. In the DMOO condition, prospective memory and planning errors may be less common because it is a more urgent cue. Yet since the DMOO cue is generally triggered later, there is an increased risk of errors due to the participant not noticing the cue in time.

We therefore hypothesize that the timing of a cue will affect the proportion of operator error types, where an earlier cue will result in more delay errors, and a later cue will result in more noticing and time-out errors. Thus, we predict that the distribution of error types will not be different when comparing an accurate DMOO invocation system with an inaccurate DMOO system, where both systems invoke the automation at the same time. Though we would expect

overall performance to be better when the automation is more accurate, based on previous research regarding the negative consequences of inaccurate automation [5, 19]. However, because the timing of DMOO and pure critical event invocation will likely differ, we hypothesize that there will be differences in the distribution of error types between these conditions. In the DMOO system, where automation is invoked later, we expect that there will be more noticing errors and time-out errors. Yet for the critical event invocation system, where automation is invoked earlier, we expect more delay errors. The reason for this is that when the DMOO invokes the automation at a later point in time there is an increased likelihood that the participant will not notice the event. However, when the critical event invocation initiates the automation at an earlier point in time, this increases the likelihood that the participant will look at the situation, decide to act on something else, but will have a prospective memory or planning failure that results in not returning to the critical event in time.

EXPERIMENT 1

To determine the role that accuracy of invocation plays in the distribution of errors in a dynamic task, we adopted Breslow et al.'s (under review) DMOO and ran it in the same simulation they used. Similar to Breslow et al. (under review) the DMOO predicted whether or not a vehicle will take damage by providing a probability that damage will occur. When the probability reached a high enough threshold, a cue was initiated. The accuracy of the automated cue was then manipulated by changing the location of the cue. In the accurate condition, the DMOO model was run in real-time and when the model predicted damage would occur, a cue flashed on the relevant hazard of the impending damage situation. In the inaccurate condition, the same DMOO model was run, but instead of flashing the cue on the relevant hazard, a random hazard (one of the 18 hazards) flashed when the model predicted damage. This resulted in approximately the same number and timing of cue invocations in both conditions, but very different levels of invocation accuracy.

METHOD

Participants

Twenty-nine George Mason University undergraduate students participated for extra credit. All participation was voluntary. Three participants' data were eliminated due to experimental error that involved an issue with the eye-tracking hardware not recording data. Four participants' data were eliminated due to low eye tracker validity. In total, 22 participants' data were analyzed. All participants had normal or corrected-to-normal vision.

Materials

The Research Environment for Supervisory Control of Heterogeneous Unmanned Vehicles (RESCHU) [21] was used in this study. A Navy pilot who is familiar with supervisory control tasks designed the RESCHU task and

the task has been used extensively. In this version of the RESCHU task, homogenous unmanned aerial vehicles (UAVs) moved on a computer screen in an environment that was dynamically changing. There were three main sections in the simulation: the map window on the right, the payload window on upper left, and a status window on the

Figure 2: RESCHU supervisory control task. The upper left panel is the payload screen that appears when a user engages a vehicle. The bottom left panel provides the user with information about each of the vehicle states. The right panel is the map view where participants must navigate vehicles to targets while also avoiding hazardous areas.
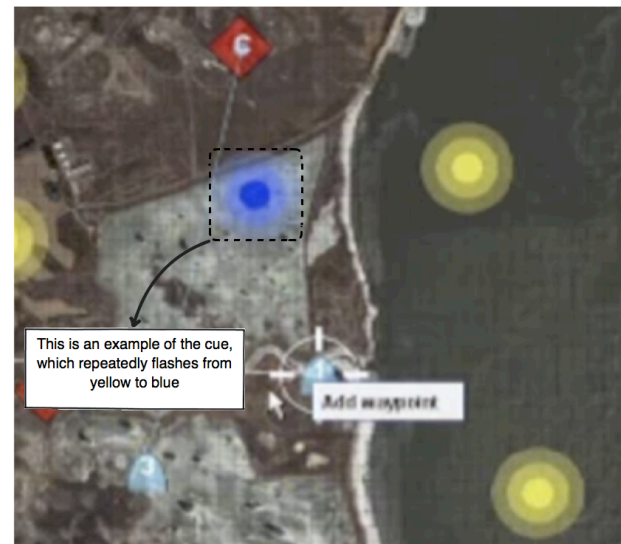

Figure 3: RESCHU supervisory control task. The circles are hazardous areas. The blue circle (highlighted with a dotted line container in order to distinguish the cue when printing in black and white) demonstrates what the cue looks like when it is fired. The half circles represent what the vehicle looks like. The red diamonds are the targets that the vehicles are directed towards.

lower left (see Figure 2). The map area displayed UAVs (blue half ovals), targets (red diamonds), which UAVs were

directed to, and hazards (yellow circles), which should be avoided. Vehicles were labeled with numbers and targets were labeled with letters. The payload window (top left) displayed a photographic image in which the participant engaged in visual search to locate an object based on written instructions as part of a payload delivery operation (described later). The status window (bottom left) depicted a timeline of each UAV's past and upcoming milestones, including the waypoints and the target of each UAV, as well as the vehicles' states (safe, damaged, dead). The simulation included five UAVs that moved at a fixed speed, 5.2 pixels per second, throughout the duration of the task. There were eighteen hazard areas, one of which changed its position randomly every four seconds, with the constraint that the hazards could not appear within three degrees of visual angle (about 50 pixels) of any UAV. If the UAV passed through a hazard, it incurred damage. Damage was indicated as a red bar in the status window. The location of targets and hazards on the simulation map was randomized with the constraint that targets and hazards were no closer than three degrees of visual angle from each other. This insured that targets and hazards could not co-occur in the same position. There were always seven targets present on the map.

Within the simulation, the system directed UAVs to targets on straight-line paths. The participant could engage targets after the vehicle arrived at a target. At the start of the simulation the UAVs were randomly assigned to targets towards which they moved along automatically generated linear paths. Once the UAV reached the target destination, the target flashed red until it was engaged. A target was engaged when the operator right clicked on the vehicle and selected the appropriate popup menu item. Engaging the vehicle triggered the payload task, where the participant performed a visual search task to locate an object such as a ship or a car in the payload window.

During the payload task, the vehicles in the map panel continued to move toward their respective targets, but operator input to the map screen was disabled. After identifying the object in the payload panel, the UAV's mission was completed. The UAV was then randomly assigned to a new target that did not already have a UAV assigned to it.

The participant also attempted to prevent vehicles from traversing hazard areas. To avoid a hazard area, the participant could assign the UAV to a different target or the participant could add waypoints to the UAV's trajectory, which effectively allowed the participant to pilot the UAV around hazard areas. The participant could also move or delete waypoints. The trajectory of the UAV was indicated with lines on the map, making it unambiguous whether a UAV would traverse a hazard area.

RESCHU requires the operator to manage multiple events that occur in parallel: more than one UAV could be waiting at a target for engagement, multiple UAVs could be on a path to a hazardous area, and it was left to the operator's discretion to act on any one of the five vehicles. In view of these task demands, the participant could not be relied on to notice whenever a UAV was on a trajectory towards a hazard.

In the accurate automation condition the equation model developed by Breslow et al. (under review) was run repeatedly (every 500 ms) in real-time with updated inputs. If the model signaled danger, a highly salient cue flashed on the hazard that the model predicted to be posing a threat (see Figure 3). In the inaccurate automation condition the equation was also run, but instead of flashing the cue on the relevant hazard to the model, a random hazard flashed. In both conditions the cue was instantiated by the yellow hazard flashing blue.

Design and Procedure

The experiment had a between groups design with the accurate automation and inaccurate automation conditions.

All participants began the experiment by completing an interactive tutorial that explained all aspects of the simulation. Participants learned about the objective of the simulation: to prevent as much damage as possible and engage as many vehicles as possible. Additionally, participants learned how to control the UAVs (changing targets, assigning/deleting/moving waypoints) and to engage a target (by right clicking on the target and selecting the engage menu item in the popup menu). Participants were also warned of the dangers of hazards and were instructed on how to avoid hazards. The tutorial lasted approximately 10 minutes.

After the tutorial, participants were instructed to practice interacting with the RESCHU simulation in the condition that they were assigned to. This practice was identical to the task that they were later be exposed to. The experimenter asked the participant to perform the actions that the participant was instructed on in the tutorial and the participant practiced the task until they were comfortable with the controls. Participants were then reminded that the goal of the task was to prevent as much damage as possible and engage as many vehicles as possible. Following this, participants were calibrated on the eye-tracker, seated approximately 66 cm from the screen, told to try to avoid damage as much as possible and to engage as many vehicles as possible, and then were administered a 10-minute session on RESCHU. This session was followed by a brief break, after which a second 10-minute RESCHU session was administered in the same manner as the first. Participants were run in the same condition for both 10-minute sessions.

Measures

Keystroke and mouse data were collected for each participant. Eye tracking data were collected using an SMI eye tracker operating at 250 hertz. A fixation was defined using the dispersion-based method, where a fixation was defined by 60 ms of sample data within a 50 pixel radius.

In order to examine how the accuracy of the automation impacted performance, the pattern of eye movements was analyzed from the moment the cue fired to when it stopped firing, which either occurred when the participant resolved the danger situation or the situation ended in damage. Fixations were categorized based on their object of focus. There were a total of five UAVs on the screen, each having a different target, and possibly different hazards associated with it. A vehicle, the vehicle's relevant hazard(s), and the vehicle's target were classified as a 'vehicle cluster'; a fixation on any of these objects was classified as a fixation on the vehicle cluster, while the initial (and possibly only) fixation on the hazard were classified as a hazard fixation.

To examine the type of errors that participants made in the accurate vs. inaccurate automation conditions, three types of errors were distinguished with respect to how participants responded to the cue: (a) Participant did not look at the situation at all (Noticing Errors), (b) Participants did not look at the situation in time to act (Time-out Errors), (c) Participants looked at the situation in time, decided to act on something else, and did not act on the cue situation in time (Delay Errors).

RESULTS

Performance on the RESCHU Simulation

A mixed ANOVA was run with condition as a between groups factor and session as a within groups factor in order to determine the impact of the accuracy of the automation on performance, where performance was evaluated based on the number of instances where a vehicle received damage by making contact with a hazardous area on the map. As expected, there were fewer instances of damage in the accurate automation condition ($M = 1.86$, $CI = .60$) than the inaccurate automation condition ($M = 3.95$, $CI = 1.16$), $F(1, 20) = 6.89$, $p < .05$, $\eta^2 = .27$. There was not a significant effect of session, $F(1, 20) = 1.12$, $p = .30$, $\eta^2 = .05$, and no interaction between condition and session, $F(1, 20) = 1.53$, $p = .23$, $\eta^2 = .07$.

The Cue's Effect on Perceptions and Actions

Recall that we hypothesized that participants would be more responsive to the cue in the accurate automation condition than the inaccurate automation condition. In support of this hypothesis, participants in the accurate automation condition were more likely to look at the cue ($M = 60.55\%$, $CI = 9.56\%$) than participants in the inaccurate automation condition ($M = 25.21\%$, $CI = 8.44\%$), $t(20) = 5.43$, $p < .05$, $d = 2.31$ (see Figure 4, first two bars). It also might be expected that participants would take longer to respond to the inaccurate automation than the accurate cue, yet there was no difference in how long it took for participants to respond to the accurate automation condition ($M = 3.79$ secs, $CI = .92$ secs) and inaccurate automation condition ($M = 4.78$ secs, $CI = 1.34$ secs), $t(20) = 1.20$, $p = .24$, $d = .51$. There was a marginal difference in the

percentage of time participants fixated on the vehicle cluster, where participants in the accurate automation condition were marginally more likely to look at the relevant vehicle cluster after the automation fired ($M = 88.83\%$, $CI = 4.07\%$) than participants in the inaccurate automation condition ($M = 82.73\%$, $CI = 4.86\%$), $t(20) = 1.89$, $p = .07$, $d = .81$ (see Figure 4, middle two bars). In further support of the hypothesis that participants are more responsive to the automation in the accurate condition than the inaccurate condition, after looking at the situation (vehicle cluster) participants were more likely to act and resolve the hazardous situation in the accurate condition ($M = 96.75\%$, $CI = 1.95\%$) than the inaccurate condition ($M = 86.57\%$, $CI = 8.63\%$), $t(20) = 2.26$, $p < .05$, $d = .96$ (see Figure 4, last two bars).

These results confirm previous findings that showed advantages of accurate automation. Additionally, the results provide a reason for why participants performed better in the accurate condition than the inaccurate condition. In the accurate condition, participants were more likely to look at the cue and were more likely to act on the relevant situation than in the inaccurate condition. This demonstrates that accurate automation results in increased responsiveness for both perceptions and actions.
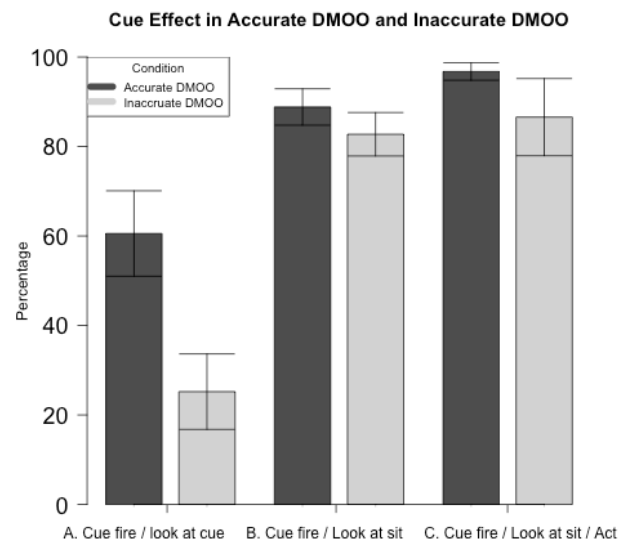

Figure 4: Perception and action behavior based on task condition. (A) The first two bars, which represent the percentage of time participants look at the cue after it fires. (B) The second two bars, which represent the percentage of time participants look at the situation after the cue fires. (C) The third two bars, which represent the percentage of time participants look at the situation after the cue fires and acted on the situation. Error bars are 95% confidence intervals.

Types of Errors

While there were fewer errors in the accurate DMOO condition than the inaccurate DMOO condition, as was demonstrated by the previous finding of improved

performance in the accurate DMOO condition, we hypothesized that the distribution of errors would be no different between the conditions. The reason for this is that in both conditions the cue was initiated using the DMOO, which fires the cue at the same point in time. The DMOO fires a cue at a critical moment, at a later point in time. Despite the accuracy of the cue, when it is fired at a later point in time this should not impact the proportion of noticing errors, delay errors, and time-out errors.

In order to explore this, instances where participants had no errors were eliminated from the analysis and a 2 X 3 mixed ANOVA was run with error type as a within subjects independent variable, and with the percentage of errors as a dependent variable. Again, condition was the between group variable. The types of errors were noticing errors, time-out errors, and delay errors. There was not a significant difference in the percentage of errors between conditions, $F(1, 16) = 1.66$, $p = .22$, $\eta^2 = .09$. There was also not a significant in the type of errors within conditions, $F(2, 32) = 1.65$, $p = .21$, $\eta^2 = .09$ and there was no interaction between condition and the type of errors, $F(2, 32) = 0.02$, $p = .98$, $\eta^2 = .00$ (see Figure 5). While there are other explanations for this non-significant effect, in Experiment 2 we will investigate whether the timing of the cue impacts the proportion of error types.

DISCUSSION
As expected, participants were more responsive to the cue in the accurate automation condition than the inaccurate automation condition. Participants looked at the accurate automation cue more often than the inaccurate automation cue and were more likely to act on the accurate automation cue than the inaccurate automation cue. Moreover, participants performed better in the accurate automation condition than the inaccurate automation condition.

In support of the idea that the timing of automation impacts the distribution of error types, there was no difference in the proportion of different types of errors that operators make in the accurate cue and inaccurate cue conditions. While more errors occurred overall in the inaccurate automation condition, there was not a significant difference in the proportion of error types in the accurate DMOO and inaccurate DMOO. We believe that there was not a significant difference in the proportion of error types in both conditions because the cue fired at the same time in both conditions, resulting in the same opportunity to respond to the automation and then decide to do something else. Thus, while the accuracy of the automation affects the number of errors and performance because accurate automation brings the participant's attention to the relevant situation, it did not significantly affect the distribution of error types. In Experiment 2 we will explicitly test the hypothesis that the timing of cue automation impacts the distribution of error types by manipulating the timing of cue invocation.

EXPERIMENT 2
To determine how automation invocation impacts operator error, and specifically how the timing of AA impacts the distribution of error types, we again used the RESCHU simulation and the same adaptive model-based cue (Breslow et al. under review). Yet in this experiment, the DMOO was compared to a simpler system that initiated automation based solely on critical events invocation of automation. In the critical event invocation of automation condition, the hazard flashed the instant that a vehicle entered a path intersecting with it. This was in contrast, to
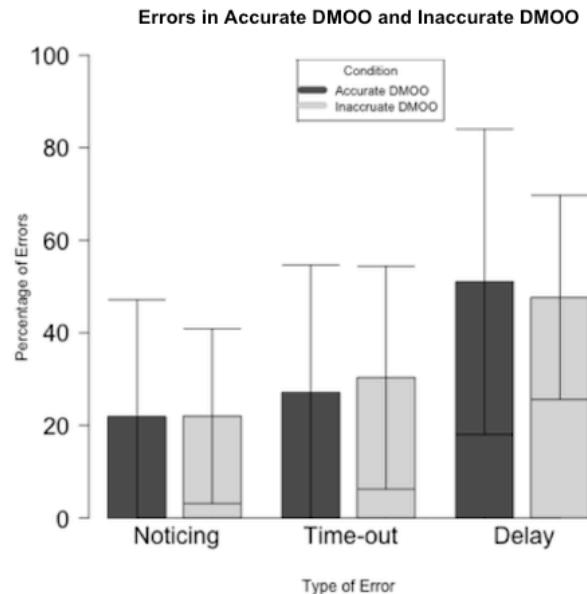

Figure 5: Type of error based on task condition. Error bars are 95% confidence intervals. A *noticing error* occurs when the participant never looks at the critical vehicle, a *time-out error* occurs when the participant looks at the vehicle, but cannot act in time, and a *delay error* occurs when a participant looks at the critical vehicle, decides to act on something else, and then does not return to the critical vehicle.

the DMOO where the cue only flashed when the system detected that there was a high probability that the operator was not going to respond to the critical event in time. This resulted in an earlier cue invocation for the critical event system condition and a relatively later cue invocation for the DMOO condition.

METHOD

Participants
Fifty-two George Mason University undergraduate students participated for extra credit. All participation was voluntary. One participants' data was eliminated due to experimental error that involved an issue with the eye-tracking hardware not recording data. Six participants' data were eliminated due to low eye tracking validity. In total, 45 participants' data were analyzed. All participants had normal or corrected-to-normal vision.

Materials

Materials were identical to the RESCHU task described in Experiment 1, with the exception that the inaccurate DMOO condition was replaced with the critical event invocation condition, which entailed flashing a cue immediately when a vehicle's path first intersected a hazardous area.

Design and Procedure

The design and procedure was identical to Experiment 1, with the exception that participants were randomly assigned to the critical event invocation condition instead of the inaccurate condition.

Measures

Measures were identical to Experiment 1.

RESULTS

Performance on the RESCHU Simulation

A mixed ANOVA was run with condition as a between groups factor and session as a within groups factor in order to determine the impact of the type of invocation on performance, where performance was evaluated based on the number of instances where a vehicle received damage by intersecting with a hazardous area on the map. There was no difference in the frequency of damage between the critical event invocation condition ($M = 2.70$, $CI = 1.11$) and the DMOO condition ($M = 3.77$, $CI = 1.77$), $F(1, 43) = 1.04$, $p = .31$, $\eta^2 = .02$. There was a significant effect of session, $F(1, 43) = 5.31$, $p < .05$, $\eta^2 = .11$, and a marginal interaction between condition and session, $F(1, 43) = 3.23$, $p = .08$, $\eta^2 = .07$.

The Cue's Effect on Perceptions and Actions

We hypothesized that participants would be equally responsive to the cue in the DMOO condition and critical event invocation condition. In line with this hypothesis, there was no difference in the percentage of time participants fixated on the cue in the DMOO condition and the critical event invocation condition, $t(43) = 0.54$, $p < .59$ (see Figure 6, first two bars). However, participants fixated on the situation a greater percentage of time in the critical event invocation condition ($M = 90.21\%$, $CI = 2.40\%$) than the DMOO condition ($M = 85.37\%$, $CI = 2.05\%$), $t(43) = 3.00$, $p < .05$, $d = 78$ (see Figure 6, middle two bars). This is likely due to the cue simply appearing for a longer amount of time in the critical event invocation condition than the DMOO condition, resulting in a greater likelihood of fixating on the situation. In the DMOO condition the cue fired on average 26.84 sec before the vehicle could potentially intersect with the hazard and in the critical event invocation condition the cue fired on average 40.92 sec before the vehicle could potentially intersect with a hazard ($p < .05$).

Also in line with the hypothesis that participants are equally responsive to the cue in the two conditions, after looking at the situation, participants were equally likely to resolve the situation in the DMOO condition ($M = 93.91\%$, $CI = $

2.61%) and the critical event invocation condition ($M = 93.72\%$, $CI = 2.88\%$), $t(43) = 0.25$, $p = .81$, $d = .03$ (see Figure 6, last two bars). These results suggest that participants had similar perceptions and actions in response to equally accurate cues.

We also hypothesized that participants in the critical event invocation condition would be more likely to look at the cue and then decide to do something else, since they had more time available to resolve the problem. As a result, participants in the critical event invocation condition should respond to the cue more slowly than participants in the DMOO condition. In support of this hypothesis, while there was no difference in how long it took for participants to look at the cue in the critical event invocation condition and DMOO condition, $t(43) = 0.34$, $p = .74$, after looking at the cue, participants took longer to act on the critical event invocation condition ($M = 7.31$ secs, $CI = .99$ secs) than the DMOO condition ($M = 5.50$ secs, $CI = .59$ secs), $t(43) = 3.11$, $p < .05$, $d = 90$ (see Figure 7). This was likely due to the fact that participants had more time to deal with the event in the critical event invocation condition than the DMOO condition.
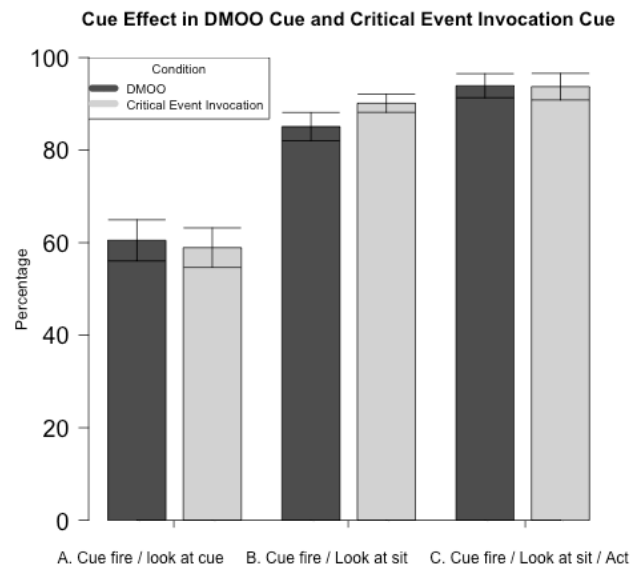

Figure 6: Perception and action behavior based on task condition. (A) The first two bars, which represent the percentage of time participants look at the cue after it fires. (B) The second two bars, which represent the percentage of time participants look at the situation after the cue fires. (C) The third two bars, which represent the percentage of time participants look at the situation after the cue fires and act on the situation. Error bars are 95% confidence intervals.

The finding that participants take longer to act in the critical event invocation condition (see Figure 7) and are more likely to look at the situation in the critical event invocation condition (see Figure 6, middle two bars) suggests that participants' strategies differ in response to the two types of

cues. While for the DMOO condition participants look at the cue and respond by quickly acting, for the critical event invocation condition, participants take longer to respond. This is due to the fact that in the critical event invocation condition there was a greater number of instances where participants looked at the cued situation and decide to resolve other problems ($M = 12.17$, $CI = 2.04$) than in the DMOO condition ($M = 5.32$, $CI = 1.56$), $t(43) = 5.27$, $p < .05$, $d = 1.51$. The reason that participant are more likely to delay dealing with the cued situation in the critical event invocation condition is likely because they have more time to return to the cued problem later.
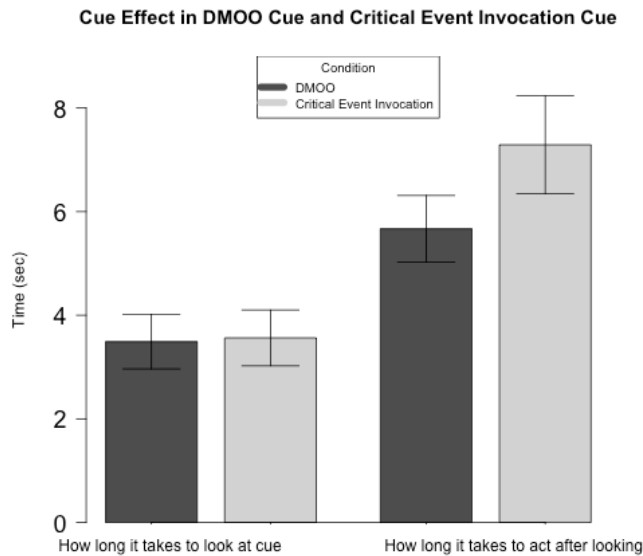

Figure 7: Timing of perception and action behavior based on task condition. Error bars are 95% confidence intervals.

Types of Errors

It was hypothesized that different proportions of error types would occur between the DMOO cue condition and the critical event invocation cue condition due to the effect of the timing of the cue. An early cue, such as the critical event invocation cue, increases the risk of delay errors, while later, urgent cues, such as the DMOO cue, increases the prospect of noticing errors and time-out errors. In order to test this, instances where participants had no errors were eliminated from the analysis and a 2 X 3 mixed ANOVA was run with the percentage of errors as a dependent variable. Condition was between group and type of error was within groups, with the type of error including noticing errors (not looking at the cue at all), time-out errors (looking at the cue but not acting in time), and delay errors (looked at something else and not returning in time). There was no effect of condition, $F(1, 33) = 0.16$, $p = .69$, $\eta^2 = .00$, and there was a main effect of type of error, $F(2, 66) = 20.02$, $p < .05$, $\eta^2 = .38$,. In support of the hypothesis that the timing of the cue results in different types of errors, there was an interaction between condition and the type of

error, $F(2, 66) = 4.69$, $p < .05$, $\eta^2 = .14$, where Benjamini Hochberg tests showed that delay errors were more prevalent in the critical event invocation condition than the DMOO condition ($p < .05$), and noticing errors were marginally more prevalent in the DMOO condition ($p = .07$). There was no difference between the conditions in the frequency of time-out errors (see Figure 8). Thus, the types of errors differ between conditions with noticing errors being more common in the DMOO condition and delay errors being more common in the critical event invocation condition. This is due to perceptual errors being more common for the DMOO invocation condition and prospective memory / planning errors being more common in the critical event invocation condition.
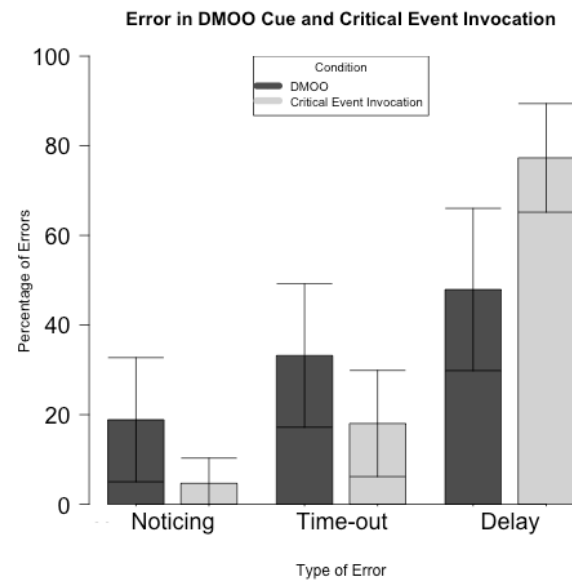

Figure 8: Type of error based on task condition. Error bars are 95% confidence intervals.

DISCUSSION

While performance between the DMOO condition and the critical event invocation condition was the same, participants responded to the cues differently and experienced different proportions of error types. For the DMOO condition, participants acted on the vehicle more quickly after they looked at the cue. This suggested that participants responded to the initializations of the cue differently, where the DMOO condition resulted in resolving the problem more quickly after it was looked at.

In support of our hypothesis regarding different types of errors based on the timing of automation invocation, in the DMOO condition, participants were more likely to make noticing errors, i.e. not looking at the cue. Yet for the critical event invocation condition, participants were more likely to make delay errors due to looking at the cue, deciding to do something else, and then not looking back at the cue. This may be due to forgetting, or not having enough time to act after returning to the problem (e.g. due

to poor planning). This suggested that a later cue, which functions as a more urgent cue, has the disadvantage of producing noticing, perceptual-based errors, while the earlier critical event invocation has the disadvantage of producing delay errors due to degraded prospective memory and poor planning.

CONCLUSION

We presented two studies that compared the DMOO with an inaccurate DMOO automation and a simpler model that invoked AA using only a critical event. The DMOO is a form of AA that invokes automation using a hybrid method that takes into account both critical events in the environment and the operator's state [14]. Of particular interest was how participants responded to the computer automation, given that automation can interact with human behavior in unanticipated ways that can have negative consequences [1, 2, 3, 4].

In support of previous research related to the benefits of more accurate automation [5, 19], participants performed better on a complex supervisory control simulation when the DMOO initiated an accurate automation than when the DMOO initiated an inaccurate automation. Participants' behavior suggested greater trust in the accurate automation, which was reflected by an increased likelihood of fixating on the accurate cue and acting on the accurate cue.

In line with of our hypothesis regarding how the timing of automation invocation affects the distribution of error types, in Experiment 1, when the automation in both conditions was invoked using the DMOO that fires a cue at an identical time, there was not a significant difference in the distribution of error types. Thus, when automation was invoked using the DMOO, participants proportionally made the same types of errors, where these types of errors included: not looking at the situation in time (noticing errors), looking at the situation but not acting in time (time-out errors), and looking at the cue, deciding to act on something else, and then not returning to the cue in time (delay errors).

Experiment 2 provided further support for the hypothesis that the timing of automation affects the distribution of error types because different proportions of error types occurred between the DMOO invocation condition and the critical event automation invocation condition. When the automation was invoked earlier, which occurred in the critical events invocation condition, participants took longer to act after looking at the cue. This suggested that a critical event invocation method would result in a greater likelihood of delay errors. Indeed, this strategy of delaying a response in the critical event invocation condition resulted in a greater proportion of errors due to the participant looking at the situation, deciding to act on something else, and not coming back to the situation in time. Yet in the DMOO condition, errors were more likely to occur due to not looking at the event in time (noticing errors).

The different proportions of error types in the critical event invocation condition and DMOO condition can be understood in terms of failures in different levels of information processing. When a cue is initiated early, there is a greater likelihood of errors due to degraded prospective memory, that is, the inability to remember actions that are planned for the future [20], or due to poor planning. Yet when automation is invoked using the DMOO method, which only alerts participants when the situation is critical, at a later point in time, participants were more likely to make perceptual errors, where they did not look at the cue before an error occurred.

The finding that the invocation of AA impacts the types of errors that operators make has implications for how to improve operator performance in various types of complex and dynamic computer tasks. Since prospective memory degrades when attention is divided [22, 23], highly complex tasks that tax working memory may benefit more from invocation of automation that occurs only at critical times, such as is the case with the DMOO. But tasks that are simpler and require that the participant be perceptually aware of the environment may benefit more from earlier cues, as in the critical event invocation condition in Experiment 2. These findings also suggest that there may be advantages to invoking multiple cues. A cue that occurs earlier could address perceptual errors and a cue that occurs at a critical moment could address prospective memory and planning errors.

One important limitation regarding the effect of timing of AA on the types of errors that operators make is that these finding may only apply to dynamic tasks where multiple events occur in parallel. For example, in a simpler task where a single event unfolds in a specific order, the operator does not have the opportunity to delay responding to a cue by addressing other components of the task.

Additionally, the finding regarding the different proportion of error types based on cue invocation may be limited to specific task parameters and levels of expertise. For example, perceived consequences can affect the operator's willingness to make delay errors because a more severe consequence for making an error may result in the operator being less likely to delay addressing the problem. Delay errors may also be less common with more heavily trained and experienced personnel because they may better understand the timing constraints of the task and therefore be less likely to make delay errors. Nonetheless, the finding regarding the different distributions of errors based on the type of cue invocation method speaks to various dynamic and time sensitive tasks where multiple events must be managed in parallel, such as air traffic control, piloting an airplane, driving an automobile, and operating a power plant. System designers must consider how the timing of automation invocation has differential effects on types of errors by invoking automation earlier to resolve perception errors and later to resolve delay errors.

REFERENCES
1. Parasuraman, R., & Riley, V. A. (1997). Humans and automation: Use, misuse, disuse, abuse. *Human Factors, 39*, 230–253.

2. Lee, J., & See, K. (2004). Trust in automation: Designing for appropriate reliance. *Human Factors, 46*, 1, 50-80.

3. Sarter, N. B., Woods, D. D., & Billings, C. E. (1997). Automation surprises. *Handbook of human factors and ergonomics, 2*, 1926-1943.

4. Sheridan, T. B., & Parasuraman, R. (2006). Human-Automation Interaction. *Reviews of Human Factors and Ergonomics, 1*, 1, 89-129.

5. Parasuraman, R. (2000). Designing automation for human use: Empirical studies and quantitative models. *Ergonomics, 43*, 931-951.

6. Parasuraman, R., & Miller, C. (2006). Delegation interfaces for human supervision of multiple unmanned vehicles: Theory, experiments, and practical applications. In N. Cooke, H. L. Pringle, H. K. Pedersen, & O. Connor (Eds). *Human Factors of Remotely Operated Vehicles. Advances in Human Performance and Cognitive Engineering*, *7*, 251-266. Oxford, UK: Elsevier.

7. Scerbo, M. (2001). Adaptive automation. In W. Karwowski (Ed.), *International Encyclopedia of Ergonomics and Human Factors*. 1077–1079. London: Taylor & Francis.

8. Opperman, R. (1994). Adaptive user support. Hillsdale, NJ: Erlbaum.

9. Parasuraman, R., Bahri, T., Deaton, J., Morrison, J., & Barnes, M. (1992). Theory and design of adaptive automation in aviation systems (Progress Rep. No. NAWCADWAR-92033-60). Warminster, PA: Naval Air Warfare Center.

10. Parasuraman, R., Mouloua, M., & Hilburn, B. **(**1999**)**. Adaptive aiding and adaptive task allocation enhance human-machine interaction. In M. W. Scerbo & M. Mouloua (Eds**).** *Automation Technology and Human Performance: Current Research and Trends*. 119-123. Mahwah, NJ: Erlbaum.

11. Kaber, D. B., & Endsley, M. (2004). The effects of level of automation and adaptive automation on human performance, situation awareness and workload in a dynamic control task. *Theoretical Issues in Ergonomics Science, 5*, 113–153.

12. Moray, N., Inagaki, T., & Itoh, M. (2000). Adaptive automation, trust, and self-confidence in fault management of time-critical tasks. *Journal of Experimental Psychology: Applied, 6*, 44–58.

13. Barnes, M., Parasuraman, R., &Cosenzo, K. (2006). Adaptive automation for military robotic systems. In *RTO-TR-HFM-078 Uninhabited military vehicles: Human factors issues in augmenting the force - NATO Tech. Rep*., 420–440. Brussels: NATO Research and Technology Organization.

14. Breslow, L., Gartenberg, G., McCurry, M., & Trafton, T. (*under review*). Dynamic Fan out: Predicting Real-time overloading of an operator supervising multiple UAVs. *IEEE*.

15. Crandall, J. W., Goodrich, M. A., Olsen, D. R., & Nielsen, C. W. (2005) Validating Human-Robot Interaction Schemes in Multi-Tasking Environments. *IEEE Transactions on Systems, Man, and Cybernetics Part-A, 35*, 4, 438-449.

16. Cummings, M. L. & Mitchell, P. J. Predicting controller capacity in supervisory control of multiple UAVs. (2008) *IEEE Systems, Man, and Cybernetics, Part A: Systems and Humans, 38*, 451-460.

17. Parasuraman, R., Sheridan, T. B., & Wickens, C. D. **(**2000**)**. A model for types and levels of human interaction with automation. *IEEE Transactions on Systems, Man, and Cybernetics. Part A: Systems and Humans, 30,* 286 **-** 297.

18. Parasuraman, R., Molloy, R., & Singh, I. L. (1993). Performance consequences of automation-induced "complacency." *International Journal of Aviation Psychology, 3*, 1, 23.

19. Parasuraman, R., & Miller, C. (2004). Trust and etiquette in high-criticality automated systems. *Communications of the Association for Computing Machinery, 47*, 4, 51-55.

20. Brondimonte, M., Einstein, G. O., & McDaniel, M. A. (eds) (1996). Prospective Memory: Theory and Application. Erlbaum: Mahwah, NJ.

21. Boussemart, Y. & Cummings, M. L. (2008). Behavioral recognition and prediction of an operator supervising multiple heterogeneous unmanned vehicles. *Humans operating unmanned systems*.

22. Marsh, R. L., & Hicks, J. L. (1998). Event-based prospective memory and executive control of working memory. *Journal of Experimental Psychology: Learning, Memory, & Cognition, 24*, 336-349.

23. McDaniel, M. A., Robinson-Riegler, B., Einstein, G. O. (1998). Prospective remembering: Perceptually driven or conceptually-driven processes? *Memory and Cognition, 26*, 121-134.